UNIVERSITY OF CAMBRIDGE
COMPUTER LABORATORY

**Technical Report No. 384**

# SECURING ATM NETWORKS

**by**

**Shaw-Cheng Chuang**

**July 1995**

University of Cambridge
Computer Laboratory
New Museums Site
Pembroke Street
Cambridge CB2 3QG
England
Telephone Cambridge (01223) 334600

# Securing ATM Networks*

Shaw-Cheng Chuang

Computer Laboratory
University of Cambridge
Cambridge CB1 3QG
United Kingdom

Email: shaw.chuang@cl.cam.ac.uk

January 1995, July 1995

**Abstract**

This is an interim report on the investigations into securing Asynchronous Transfer Mode (ATM) networks. We look at the challenge in providing such a secure ATM network and identify the important issues in achieving such goal. In this paper, we discuss the issues and problems involved and outline some techniques to solving these problems. The network environment is first examined and we also consider the correct placement of security mechanism in such an environment. Following the analysis of the security requirement, we introduce and describe a key agile cryptographic device for ATM. The protection of the ATM data plane is extremely important to provide data confidentiality and data integrity. Techniques in providing synchronisation, dynamic key change, dynamic initialisation vector change and Message Authentication Code on ATM data, are also being considered. Next, we discuss the corresponding control functions. A few key exchange protocols are given as possible candidates for the establishment of the session key. The impact of such key exchange protocols on the design of an ATM signalling protocol has also been examined and security extension to an existing signalling protocol being discussed. We also talk about securing other control plane functions such as NNI routing, Inter-Domain Policy Routing, authorisation and auditing, firewall and intrusion detection, Byzantine robustness. Management plane functions are also being looked at, with discussions on bootstrapping, authenticated neighbour discovery, ILMI Security, PVC security, VPI security and ATM Forum management model.

*A shorter version of this paper has being submitted in July 1995 to The 3rd ACM Conference on Computer and Communication Security

# 1 Introduction

Broadband ISDN (BISDN) has been conceived as a universal network supporting different kind of applications and customer categories. It enables the provision of a wide range of services to a broad variety of users utilising a limited set of connection types and multipurpose user network interfaces. Asynchronous Transfer Mode (ATM) has been identified as the transfer mode [1] for implementing BISDN. In recent years, much interest has been expressed in Asynchronous Transfer Mode (ATM), due to its flexibility and its support of multimedia traffic.

Such interest has lead to some preliminary deployments and intense research in issues such as congestion control, quality of service provision etc. However, the concerns of providing a secure ATM network service has been largely ignored. It is envisage that such ignorance of security issues can be a hindrance to future wide-scale deployment of ATM services. This is particularly important in the race towards the building of the "Information Superhighway". With a wide spread communication infrastructure, previously isolated organisations are now being interconnected. The correct handling of security issues such as access control, data confidentiality/integrity etc is essential to safeguard the operation of such an organisation.

In the past, security issues has been handled with various ad hoc and fire-fighting approaches, often after a network service has been fully deployed. Security service is often "bolted" on as an afterthought. As a result, the outcome of such work tends not to be very satisfactory. In our work on security service for ATM networks, we try to avoid such pitfalls.

In this paper, we look at the challenge in providing a secure ATM network. We first examine the network environment that we are protecting and consider the correct placement of security mechanism in such an environment.

We then proceed with examining the important issues involved in securing ATM networks and outline some techniques to solving these problems.

The concept of a *CryptoNode*, a key agile cryptographic device for ATM, is introduced with discussion on the issues involved in the construction of such a device.

The reference model for the BISDN protocol stack, as shown in Figure 1, identifies three protocol plane, the user, control and management planes. This clearly distinguishes the different functions required for data transfer, network control and network management. In this paper, we use the term ATM and BISDN interchangeable and organise our discussion around these three planes. The reader of this paper is expected to have an understanding of the concept behind ATM network.

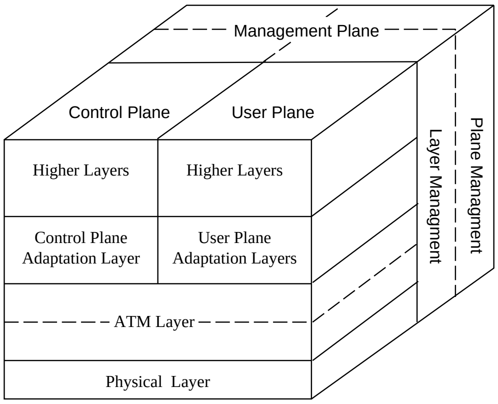


Figure 1: BISDN Reference Model

The protection of the ATM user plane is extremely important to provide data confidentiality and

[1]Transfer mode is a specified way of transmitting and switching information in a network

data integrity. Techniques in providing synchronisation, dynamic key change, dynamic initialisation vector change and Message Authentication Code on ATM data are also being considered.

Next, we discuss the corresponding control functions. A few key exchange protocols are given as possible candidates for the establishment of the session key. The impact of such key exchange protocols on the design of an ATM signalling protocol has also been examined and security extension to an existing signalling protocol being discussed. We also talk about securing other control plane functions such as NNI routing, Inter-Domain Policy Routing, authorisation and auditing, firewall and intrusion detection, Byzantine robustness.

Management plane functions are also being looked at, with discussions on bootstrapping, authenticated neighbour discovery, ILMI Security, PVC security, VPI security and ATM Forum management model.

## 2 Network Environment

Before we can to consider the security threats to the ATM networks, we need to consider the environment we are protecting. We are looking at an environment where native ATM is used everywhere, as it is described in [1]. This is illustrated in Figure 2.

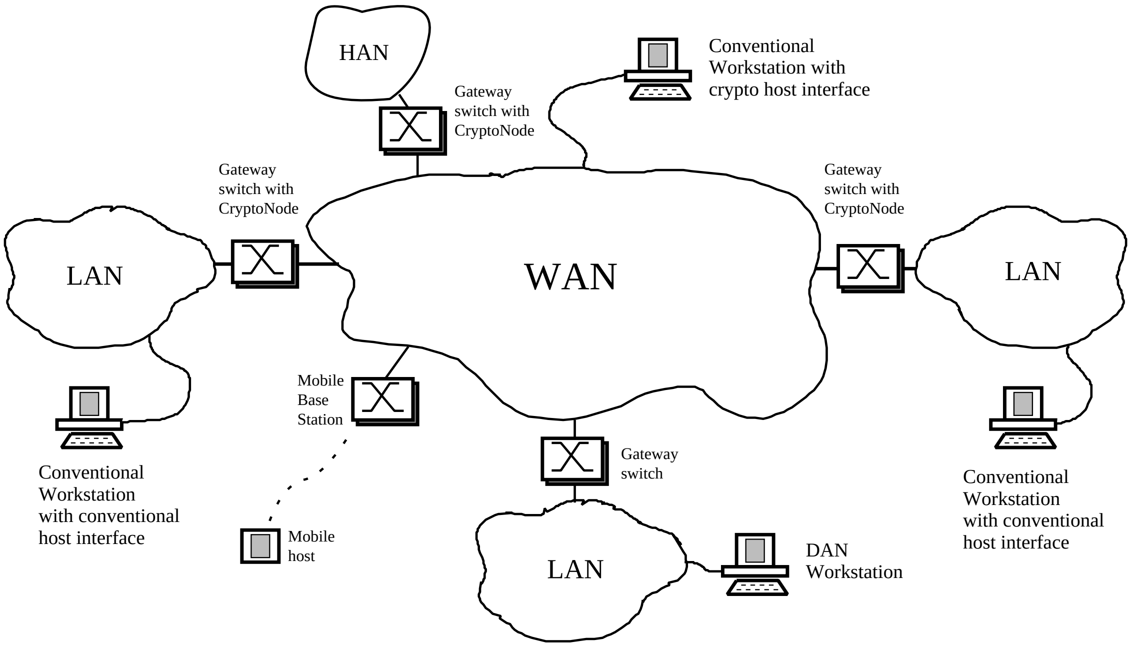


Figure 2: ATM Network Environment

In recent years, the deployment of ATM network equipment has gathered pace. One of the most promising areas for the introduction of an ATM-based network is ATM LANs, which are also known as third generation LANs. Such ATM LANs are then interconnected using either private ATM WANs or public ATM WANs. Equipment that are attached to these ATM LANs and ATM WANs include:

- Workstations with ATM interfaces

  There are lot of these already in deployment. Therefore security mechanisms compatible to this is highly desirable.

- Workstations with cryptographic-capable ATM interfaces
- ATM devices such as ATM Cameras [2], ATM Disk etc
- Multimedia workstations such those based on Desk Area Network (DAN) [3]
- Home ATM appliances built around the Home Area Network (HAN)
- Wireless ATM devices [2]

Private ATM LANs are sometime considered to be locally secure but traffic across the interconnected ATM WANs are however considered vulnerable and required protection. Therefore the gateway switch between the ATM LAN and the ATM WAN can be treated as a security firewall. As a result it is desirable that the gateway switch will possess the necessary hardware to performs the necessary cryptographic functions. Independent research at MCNC [4, 5] is based on a similar viewpoint.

The granularity of protection at this firewall normally will fall into the categories of application layer firewall or network layer firewall [6]. Tennenhouse [7] has described the principle of avoiding unnecessary multiplexing in the ATM environment that we are considering. Much benefit in terms of jitter, performance and Quality of Service (QoS) cross-talk can be gained from minimum multiplexing. Therefore we are advocating that if an application does not multiplex over the same virtual circuit, then there is literally no difference between application layer firewall and network firewall. This view of providing firewalling reduce our problem to the case of providing a general solution of per-VC protection.

# 3 Challenges in Securing ATM

There is so far little, if not none, security considerations in the current standardised work on ATM. Therefore a wide range of functionalities in ATM require security considerations and protections. Some part of the new security measures required also apply to other network technologies. However, the high speed cell relay nature the ATM networks present some unique problems to the task of securing it. In this section, we present a discussion of these unique issues. Later on in this paper, we will then provide a more detail discussion on some more issues and propose solutions to some of the problems.

## 3.1 High Speed Networking

The potential high speed service of ATM networks creates a problem for the security management. Due to the high data volume, any session key lifetime becomes very short in term of time duration. For example, if we consider a compressed High Definition Television transmission at rate of 130 Mb/s, the key lifetime is in the order of 10 seconds (assuming that a key is never used for more than $10^9$ cipher blocks). Thus we require to have a mechanism whereby a session key could be changed very rapidly during the lifetime of a call. The traditional full key exchange protocol seems unsuitable in this context. Since it is also possible to have a long lifetime connection, mechanisms are also expected for the renewal of the session key on a slightly longer term basis (e.g. daily).

## 3.2 Fine Grain Multiplexing

Fine grain multiplexing at the cell level is the key feature of ATM and this could present a serious challenge to the designer of a high speed crypto-system for ATM. Security operation for ATM can be performed at various levels, namely at:

[2] To be discussed in details in next release of this document

- Cell stream level with both header and user payload protected

  This is equivalent to link layer protection and it has the additional advantage of preventing traffic analysis. However this is rather inflexible since it works almost entirely on a link by link basis as ATM switches would require to operate on the cell header. Since ATM cells are switched at any intermediate ATM switches, cell payload needs to be decrypted at one end and re-encrypted at the other end, each with a different key. Thus not only do we have two encryptions at each switching point, we also have security exposure at the intermediate switch. Of course, we can avoid the double encryption by assigning the same key for all traffic in the entire network. This is however clearly not a secure option.

- Cell stream level with user payload protected

  ATM cell's payload are protected while in transit between two ATM endpoints with a single key. However user data is protected under a single security level which is clearly not sufficient.

- Per-virtual circuit (VC) level with user payload protected

  ATM offers the concept of a VC which is operating on an end-to-end basis. If applications do not multiplex their traffic on the same VC, then a VC is equivalent to a secure channel in the security context. Security policy which reflects application requirement can be applied on a per-VC basis. Therefore it is desirable to have per-VC user payload protected, i.e. each VC has a unique key.

### 3.3 Selection of Cryptographic Mechanisms

ATM has been seen as a scalable network service where the operating speed can range from tens of Mbps to Gbps. When operating at the high end of the speed range, this is normally beyond the capability of traditional cryptographic mechanisms. Most of the traditional cryptographic mechanisms operates up to ten of Mbps range when implemented in software and up to a few hundred Mbps when implemented in hardware. Even though the state-of-the-art high speed encryption device [8] operates at Gbps, it works under the assumption that all data is protected under the same key and there is a big overhead to change the cryptographic context. Due to the key agile requirement as discussion above, this is not the ideal candidate for securing ATM.

The other factor that affects the selection of cryptographic mechanisms is the small ATM cell size. The ATM cell payload is 48 bytes and this restricts us to block cipher that operates on block size less than 384 bits, if we want to avoid getting into the complication of cross-cell cryptographic operations. This rules out algorithm such as RSA [9]. For efficiency reason, we also like to have alignment of data in the cell with the cryptographic block.

### 3.4 Support for Open Multi Service Networks

It is envisaged that future development of ATM networks is towards *Open Multi Service Network (OMSN)* [10], which allow third party software to be used as well as allow third party to provide services. The creation of an OMSN will open up new possibilities of use and abuse. One possible approach to produce an *Open Service Support Architecture (OSSA)* in an OMSN as identified in [11].

### 3.5 Secure ATM Charging

ATM's ability to support multi-service traffic complicate the accounting and auditing practice due to the potentially large number of traffic classes. The task of finding a suitable tariffing scheme or economic model for running the ATM network is an interesting challenge. How this interacts with the upcoming electronic monetary system remains an unresolved problem.

# 4 Security Mechanism Placement

In the following sections, we will discuss the placement of security mechanisms on different ATM operating environments for the efficient execution of per-VC protection. The selection of the operating environment and security mechanism is very much dependent on the application requirement.

- Conventional Workstation

  Traditional workstations are being equipped with ATM interfaces with the advent of ATM networks. Thus any security mechanism which is inter-workable with the existing workstation platform will be beneficial to the integration and wide spread use of security mechanisms. Since AAL5 is a very widely implemented ATM Adaptation Layer (AAL), we will use this as the primary example, though many of the concepts are applicable to other AALs. Operating at a suitable AAL abstraction not only allows the security function to fit in with the application data unit [12], it also increase network efficiency and simplify the operations of implementing the security function. This is particularly true when we are dealing with data integrity as we shall see in later sections. The security function on the workstation platform can be implemented at three levels:

  - Host interface hardware

    With some high performance and intelligent ATM host interface hardware, it is possible to integrate the cryptographic hardware with relatively little extra production cost. Ideally, the cryptographic function on the VC stream should be performed near where the Cyclic Redundancy Code (CRC) function is performed and near the IO bus of the workstation. With this design, processing latency can be greatly reduced with cut through of various degrees. Of course, device driver support is also required. The degree of cut-through is highly dependent on the buffer management scheme at the host interface.

  - Device driver

    A very cost effective way to implement security function is to implement it in the device driver. This is particularly true for relatively simple ATM host interfaces where most intelligence are placed in the device driver. With ever improving microprocessor processing power, this might be a very attractive option for achieving higher crypto performance. Implementing in the device driver provides an advantage that it is totally transparent and provides easy integration to existing platforms and applications.

  - Secure middleware

    It is of course possible to take the security function totally outside the lower level device driver and hardware interface. A secure middleware could be implemented "on top" of a VC. A security related data unit is passed transparently from the device driver and is processed by this layer of middleware before being handed over to the application. However, in this case there is a danger that the key used in the cryptographic operations can be exposed to the application.

  The detailed integration of security interface within an end system operating system, such as Unix and multi-service capable system like Nemesis [13] and the DAN Operating System, is the subject for another paper. Issue like the suitability of generic cryptographic service API such as the X/Open GCS-API [14], GSSAPI [15] and the NIST API [16] in such an environment will also be discussed.

- Desk Area Network

  The concept of a Desk Area Network (DAN) [3] uses the ATM interconnect to provide the interconnections between the components of the machine and from these components

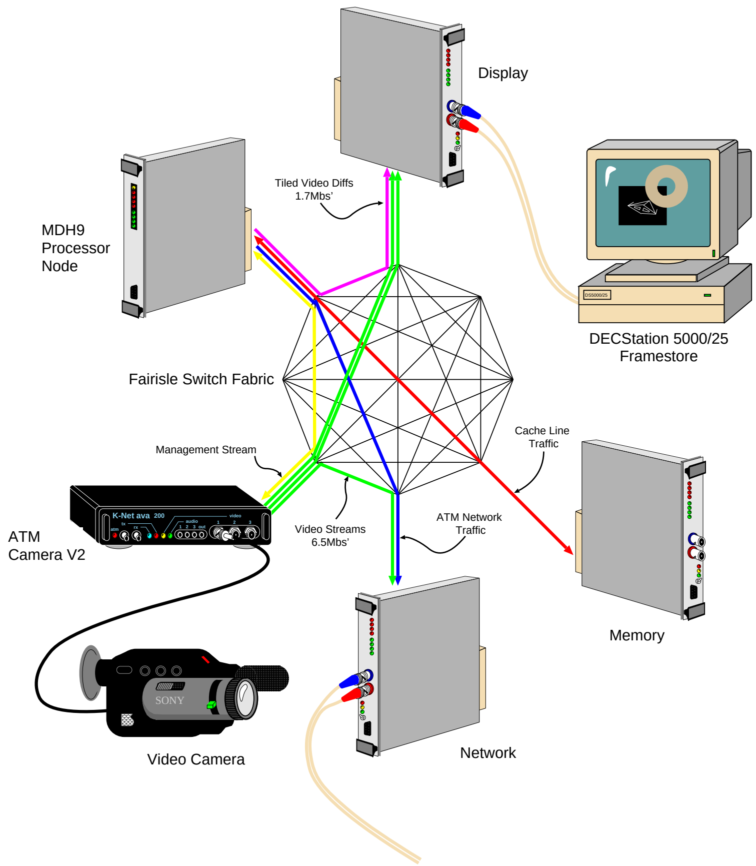


Figure 3: DAN - The Desk Area Network

directly to a high speed ATM network. However, all components are treated as would be the components of a workstation. This is as shown in Figure 3.

In a DAN environment, all the devices attached to the same interconnect are considered to be within the same security domain, under the control of the DAN operating system (DANOS). Hence it is natural to consider the DAN network interface as the security boundary. If secure communication is desired, security policy must be enforced at (or before) this point. The control of this cipher-capable secure DAN network interface within the DAN, by the DANOS, can utilise proprietary protocols. However, the ciphering device must also speak the User-Network Interface protocol to interface to the outside world. As far as the network is concerned, it is considered to be a communication endpoint.

Also unlike the normal workstation environment, we have different types of DAN devices generating/sinking traffic. These DAN devices have different degrees of intelligence in terms of control processing power. It is also a major cost factor if a security mechanism has to be implemented in all these devices. The placement of ciphering hardware at the network interface allows it to be shared amongst the multiple components present on the DAN. In particular, with suitable DANOS control, we can utilise the ciphering device as a secure coprocessor for the DAN as well. Consequently the secure DAN network interface needs to possess a certain level of intelligence to carry out the necessary processing. of course, in this model, we do not preclude the possibility of having crypto-capable DAN devices. The choice of crypto-capability placement in this case, and also in the case of conventional workstation, is between the complexity/cost of building cryptographic mechanism and having a simple point of crypto processing. Implementation of cryptographic mechanisms normally involve some kind of tamper-proof mechanisms which could be both complex and costly. However, if we concentrate all our cryptographic processing to a single node, we push up the cryptographic processing speed requirement. High speed cryptographic mechanisms, with the additional key agile requirement of ATM also complicates the design of such device.

In the DAN environment, we are expected to handle multimedia high speed communication, which generate large data volume. In this situation, the issue of key lifetime become particularly important. The issue of key agility due to the effect of cell stream multiplexing is less severe here since we are expecting a slightly lower level of multiplexing than in a normal ATM switch environment.

The Home Area Network (HAN) environment is considered to be very similar to the DAN, with low cost ATM appliances connection within the protection boundary of interface to the external world.

- Switch node

  In the case of a switched firewall, the switch at the boundary needs to handle a high level of cell stream multiplexing. Of course, the control issue is very different. The DAN network interface is considered to be an endpoint whereas the switched firewall is considered to be a network node which result in different handling of signalling protocol. Nevertheless this can be handled in a higher level by the general purpose processor. Other than this, the requirement on such a device seems to be very similar to that of secure DAN network interface.

## 4.1 CryptoNode - A Key Agile Cryptographic Device for ATM

From the above discussion and analysis of ATM networks security, especially those on DAN and secure ATM switch, we bring out the concept of a *CryptoNode* - a key agile cryptographic device for ATM. As we have shown in the previous section, both the DAN secure network interface device and the secure ATM switch port has very similar specifications:

- Interface with the switch fabric directly
- Possess general processing intelligence
- Perform high speed encryption on cell streams
- Allow per-VCI encryption as described in Section 3.2.
- Preserve the Quality of Service (QoS) of the data it processes
- Acting as security filter, i.e. the flow of traffic must not only from fabric to output or vice versa, we must also allow fabric to fabric processing.

It is envisaged that such a device can also be used in a more general purpose manner. One suggested use is in building DAN-based Internet Protocol (IP) router. In such environment, as shown in Figure 4, the *CryptoNode* is used as the general secure processing unit for processing of higher level protocols such as IP. In such a router, the encryption module (*CryptoNode*) is shared among all the different network interfaces. This can be a very cost effective way of constructing a router, rather than building cryptographic capability into every network interfaces. This is particularly true if not all traffic need to be protected.

The same argument can be applied in an ATM switch environment. Rather than having a *CryptoNode* in every interface, we can have a few *CryptoNodes* in a switch. More than one *CryptoNode* is used mainly for load balancing purposes and the number of *CryptoNodes* will depend on the percentage of secure traffic. The mode of operation for ports without *CryptoNodes* is to route the

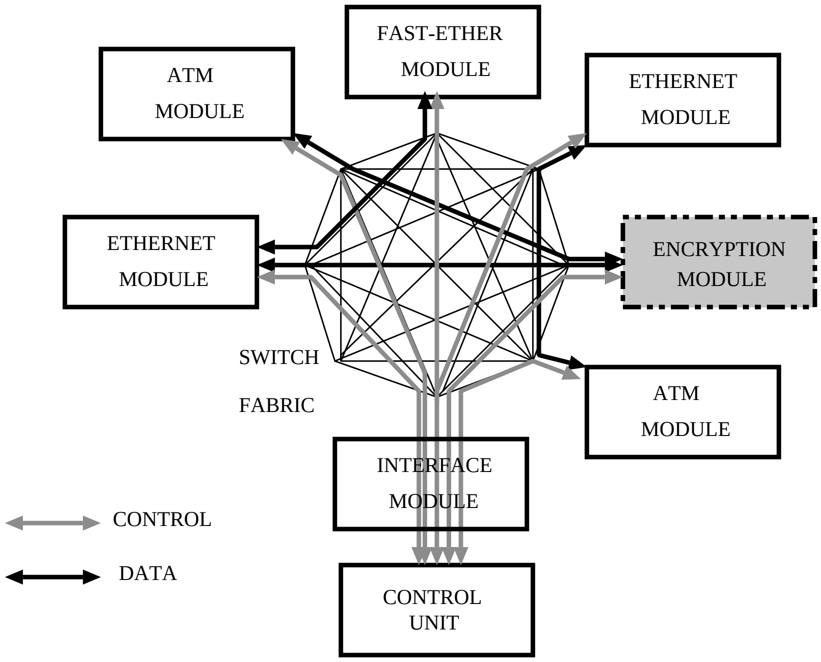


Figure 4: DAN-based IP Router

secure traffic via one of the *CryptoNode* before going to the output port, i.e. the *CryptoNode* is acting as a security filter. One implication of this arrangement is that we need to increase the fabric speed, to account for the extra hop across the fabric. This has also an impact on the QoS of the traffic since this imposes extra delay. Therefore such configuration presents a tradeoff between performance and cost.

#### 4.1.1 Structure of a *CryptoNode*

In Figure 5, a block diagram of a design of the *CryptoNode* is shown. The *CryptoNode* consists of mainly three parts:

- The network part DMA engine

  This section consists of static memory, buffer FIFO and the xilinx control chip. The static memory is used for cell buffering, and resides in the memory mapped I/O space of the processor. The xilinx arbitrates access to this memory between the processor, crypto part, VCI mapping, transmission to the fabric and reception from the network. The FIFOs are between the transmission system and the rest of the system to decouple the different transfer speeds.

- Processor part

  This section has a processor such as an ARM processor and runs the control operating system. The section also has dynamic memory and ROM. The interface to the expansion bus allows addition of other network interfaces.

- Crypto part

  This sections performs the high speed ciphering operations. All streams that require cryptographic operations will be directed by the network part to be processed by the crypto part. This sections consists of high speed encryption devices, xlnix control chip, random number

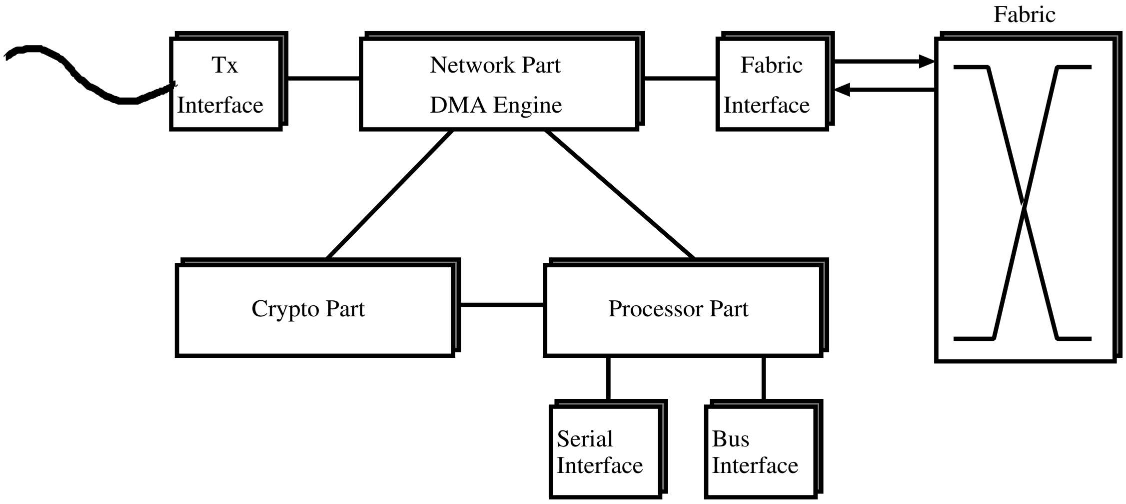


Figure 5: Structure of CryptoNode

generator and static memory. The xilinx performs the detection of *CryptoTag* (see Section 5.3.1), controls the operations of the high speed encryption devices such as the loading of ciphering context and general error recoveries such as recoveries from synchronisation loss.

### 4.1.2 High Speed Encryption Devices

It is worth considering in more detail the selection of high speed encryption devices. Most of the so-called high speed encryption devices achieve their throughput while operating under the same cryptographic context, i.e. under the same key with the same stream. Any change of cryptographic context imposes a big penalty on performance as mostly it involves halting the ciphering operation and going into a special context loading mode. This is however unacceptable in the case of per-VC ciphering for ATM traffic. Problems of similar nature have also been observed with other kinds of processing, such as JPEG, of multiplexed ATM streams.

There are a few possible approaches that can alleviate this problem:

- Parallelised operations

  We can design parallelised ciphering modes so that we can reduce the speed requirement on the encryption devices. For example, an ATM cell payload is 6 Data Encryption Standard (DES) blocks and it is therefore possible to design a parallelised DES Cipher Chain Block (CBC) mode that in effect split the single stream into 6 pseudo streams to be processed by 6 DES devices individually. This approach allows slower and so cheaper devices to be used. Nevertheless it still does not completely solve the problem of rapid loading of cryptographic context.

- Improved cache

  Some encryption devices allows a very limited amount of cryptographic context to be cached with the device. An enlargement of this cache can reduce the number of context missed and hence improve the overall performance. This improvement in performance is however very much dependent on the traffic characteristic of the ATM traffic. With large degree of multiplexing at points such as the ATM WAN, the gain might not be significant.

- General purpose processor with flexible cache

In recent years, advances in the software implementation of cryptography algorithm such as DES has greatly improved the performance. Furthermore, the performance of general purpose processors has greatly improved to the stage that it might be feasible to utilise such processors to performance specialised operations such as cryptography. This will still have the problem of cache miss on the cryptographic context. The consideration of multimedia traffic processing in the context of the DAN work, also resulted in the proposal of flexible cache architecture for QoS guarantee stream processing [17, 18]. One feature of such an architecture is the ability to "lock" certain data into the cache so that the penalty of a fetch from DRAM is never incurred. Hence in our application, if the cryptographic context is locked into the cache and if the cache is sufficiently large, we have an almost equivalent of the improved cache case described above. Of course, the same drawback still exist. We however gain the advantage of having a very general and flexible solution.

- Wide interface encryption device

  One natural approach to solve the cryptographic context loading problem is to allow all cryptographic context to be loading simultaneously with data. This has the consequence that the interface to the encryption device needs to be extremely wide and the implication that the cost of producing such device and the corresponding system will be much higher.

# 5 User Plane Protection

In this section we will discuss a few issues involved in the protection of the ATM user plane.

## 5.1 Threats

The threat to the user plane includes active or passive intruders capturing, modifying or replaying any data on the ATM network. The countermeasure that we are considering includes both data confidentiality and data origin authentication. Independent work in [4, 5] has considered only the importance of data confidentiality. While this is important, maybe more so in a military style environment, it is felt that the guarantee of data origin authentication can be equally or even more useful.

The provision of data confidentiality involve the encryption of sensitive data in such a way that it may only be correctly deciphered by the intended parties.

The provision of data origin authentication can be achieved using a Message Authentication Code (MAC). With a MAC, we can be sure that the message that we received is originated from the intended party and the data integrity is ensured.

When ATM networks is used with a non-discretionary security policy, the resulting system might possess covert channels. The existence of covert channel poses a potential threat to the use of the mandatory policy throughout the system's life cycle. Thus we attempt to eliminate the covert channels or at least reduce the bandwidth of the covert channels.

## 5.2 Granularity of protection

This issue has been considered in detailed in Section 3.2.

## 5.3 Synchronisation points

ATM networks are expected to lose cells. For a lot of cryptographic schemes, data loss will imply the loss of synchronisation and result in the incorrect processing of subsequent data, i.e. the error propagates/extends. We can propose the cryptographic schemes to be used in the ATM networks to be free of property of error propagation, this is however a very restricted solution. In our design to protect the ATM data path, we aim to have the flexibility of allowing arbitrary cryptographic schemes to be used which bring us to consider the synchronisation point problem in more detail.

Synchronisation points are tokens inserted into the ATM data stream for the purpose of re-synchronising the state of the deciphering end with the enciphering end. The synchronisation point needs to be inserted in-band (to the same VC) since while ATM guarantees cell order within a VC, it does not guarantee cell order between different VCs. For the moment we consider the synchronisation point to be of arbitrary size. Synchronisation points are also important in the case of providing data origin authentication. The attachment of a MAC to the data stream for the purpose of integrity (we use the term integrity and data origin authentication interchangeable in this paper) is in a way a kind of synchronisation point. Thus in the consideration of placement and structure of synchronisation point, we need to consider very carefully the coverage of this MAC.

There are a few approaches to the insertion of synchronisation points in the ATM data stream:

1. Periodic

   For ATM cell streams, such as those from Constant Bit Rate (CBR) traffic, synchronisation points can be at periodic intervals. The period of the insertion can be determined from the data rate of the stream. In many system, the problem is not so much preventing denial of service, but in detecting it. The periodic synchronisation tag can also serves as the liveness test of the virtual circuit, thus countering denial of service attack on the VC. This approach is however not suitable for Variable Bit Rate (VBR) traffic or Available Bit Rate (ABR) traffic.

2. Encapsulation

   ATM Adaptation Layer (AAL) lies between the ATM layer and higher layer. Its basic function is the enhanced adaptation of services provided by the ATM layer to the requirement of higher layer.

   One possible approach to handle synchronisation is to encapsulate a synchronised block within an AAL unit. This could be either a AAL5 frame or a unit of a new "Secure AAL". While this approach has the advantage that other cryptographic information can be encapsulated together, it has the problem that:

   (a) If AAL5 is used, AAL5 frame length might exceed 64K bytes and therefore fragmentation is needed. This will complicates the protocol processing.

   (b) If a new Secure AAL is used, it will not be compatible with most of the ATM interfaces.

3. PDU boundaries

   We want to carry out integrity check at suitable PDU boundaries so that the application can efficiently discard corrupted data unit. Applications of ATM networks interface with the AAL service and do not see the ATM layer. Therefore the suitable PDU boundary will be the AAL frames boundary. Based on this argument, we can conclude that we want a per-VC level user payload integrity protection with awareness of AAL framing structure.

   This approach violates the principle of layering where lower layer should not require to understand higher layer structure. However, experience in the networking community has shown that such approach can lead to great improvement in efficiency. The efficient usage of the ATM network and end system handling of data are very much dependent on the selection of the right AAL. It is clear, e.g. from the experience of running TCP over ATM [19], that selective operation at the ATM level using information from the AAL can highly optimise communication efficiency. Applying this approach to every conceivable AAL types is however not considered possible. Since AAL5 is the most widely used AAL, it is worth paying special attention to its treatment.

From the above discussion, we conclude that our method of synchronisation will be a combination of periodic insertion and insertion at PDU boundaries. For non AAL5 traffic, only periodic insertion will be used. However, for AAL5 traffic, synchronisation points are inserted normally at AAL5 frame boundaries. During traffic idle time, periodic insertion of synchronisation points also serves the purpose of liveness test.

### 5.3.1 Structure of the synchronisation token

As we have indicated in the above section, the synchronisation token, which we call the *CryptoTag* from now on, must be inserted in-band. There are a few ways that we can achieve this:

- Operations and Maintenance (OAM) cell

  New type of OAM cell type can be defined for the purpose of synchronisation. Such approach however will not be compatible with existing ATM hardware platforms and this particular usage of is not really for "Operations and Maintenance".

- Cell with special payload pattern

  Special bit pattern in the payload of a ATM cell is used to recognise the arrival of a *CryptoTag*. However for cell payload that contain this special bit pattern, they will be mis-identified as the *CryptoTag*. As a result this approach can in fact worsen the synchronisation problem by generating false synchronisation points.

- Special AAL5 frame

  This approach constructs a *CryptoTag* using a Special AAL5 frame with the the *CPI* field set to a special reserved value. We have stressed in the previous sections that one of the important issues is to provide a mechanism that is compatible with current ATM hardware platforms. The *CryptoTag* in this case can still be easily detected by hardwares that already deal with AAL5. Given that it is totally compatible with AAL5, the processing of such *CryptoTag* in the device driver or the secure middleware is not a problem. In the case of secure middleware, one extra requirement on the lower layer software is to deliver the *CPI* field to the application. We have adopted this as the *CryptoTag* in our system.

With the basic structure of the *CryptoTag* determined, we need to decide on the size of such *CryptoTag* and the kind of information that it carries.

We have decided that a single cell AAL5 frame is the best engineering tradeoff for the *CryptoTag*. The tradeoff is between ease of detection, network efficiency, buffer requirement and the amount of information it carries. *CryptoTag* does not carry user information, a bigger *CryptoTag* therefore reduces user data throughput. Furthermore, since it is a valid AAL5 frame, and the *CryptoTag* identifier is in the AAL5 trailer, the whole *CryptoTag* needs to be buffered until the end is detected. Beside the extra buffering memory required, any buffer at the *CryptoNode* or equivalent device increases the delay, and hence worsen the QoS performance.

With a single cell size *CryptoTag*, the detection algorithm can be very simple:

> If the bit (indicating the last cell of the AAL5 frame) is set and the length of the cell is one cell and the *CPI* field is set, then this is a *CryptoTag*.

### 5.3.2 *CryptoTag* content

Since the AAL5 trailer is 8 bytes long, we are left with 40 bytes in the *CryptoTag* for carrying cryptographic information. The information content is structured as shown in Figure 6.

| | |
|---|---|
| Key Hint | 64 bits |
| Confidentiality IV | 64 bits |
| Integrity IV | 64 bits |
| MAC | 128bits |

Figure 6: Structure of *CryptoTag*

- Key Hint

  This field indicates that there is a key change event and is interpreted by the higher level software. This permits a lot of flexibility on how the field should be interpreted. Indeed, the exact interpretation can be determined at call setup time as part of the negotiated security profile. One naïve way to use the key hint field is to put an encrypted key in the field. Another possible way the key hint field can structured is as shown in Figure 7:

  | | | | |
  |---|---|---|---|
  | Confidentiality Master Key | Confidentiality Session Key | Integrity Master Key | Integrity Session Key |

  Figure 7: Structure of *Key Hint* field

  As we can see, there are the *Master Key* and *Session Key* fields for both confidentiality and integrity. In this case, we are assuming that there is a longer-term *Master Key* and a short term working *Session Key* for ciphering of data. Each of these field can be used as an index into a table of keys. Alternatively the real *Master Key* can use the *Session Key* field to generate the real *Session Key*, e.g. by encrypting ($K_M$, K) where $K_M$ is the master key.

  The *Master Key* is changed on a much longer time scale, e.g. daily. Therefore we can employ the normal full key exchange procedure. The *Master Key* field change is then used as a trigger for the new *Master Key* to be registered.

  The change in *Key Hint* field can be easily implemented in the hardware by a simple comparison of current *Key Hint* with the previous *Key Hint*. If a change is detected, interrupt to the processor is generated. The cell level processing by the low level hardware is halted and the control of processing is passed to higher level software. When the processor has finished with the interpretation of the *Key Hint* field, the virtual circuit is scheduled to be restarted. Thus we required per-VCI queueing of the cell streams.

- Confidentiality IV and Integrity IV

  The Confidentiality IV field is the Initialisation Vector (IV) used for the encryption of the cell payloads. Notice that this field is only 64 bits. It might seems that we are limiting ourselves to the case of 64 bits IV block size. However, we should also observed that no IV is required in the calculation of many MACs. Consequently it is possible to reuse the Integrity IV field to extend the Confidentiality IV field to 128 bits.

- MAC

  This is the MAC to provide the data origin authentication. The coverage of the MAC is as shown in Figure 8:

  Notice that MAC coverage end on a 8 bytes boundary. It is carefully arranged so that the structure allows the validation of the MAC to be performed in a pipeline fashion.

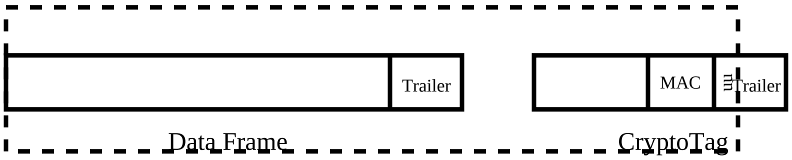


Figure 8: MAC Coverage

There is no dependency between the confidentiality and integrity function. The selection of which function to be used for a particular VC is determined at call setup time by specifying the required security profile.

It should be noted that the *Key Hint*, *Confidentiality IV* and *Integrity IV* apply to the next block of data, while the MAC corresponds to the current block of data. The *CryptoTag* is always sent back-to-back with the current block of data. By sending the context information before the generation of next data block, we allows the receiving end to prepare for the possible change in context. This can reduce the amount of stoppages in the data stream and therefore presents a better QoS.

Notice that we do not have a sequence number in the *CryptoTag*. As a result we are not able to detect message replay. Besides being an engineering tradeoff, due to the lack of space in a single cell, it was also felt that replay detection can be better handle by the higher layer application. The ATM layer should not be loaded with too much state for each VC. Furthermore, for many applications such as a reliable transport protocol based on sequence number, replay detection is intrinsic and we should not waste resources in the lower layer duplicating the effort.

## 5.4 *CryptoTag* Compatability and Extensibility

If there is hardware support for native AAL1 and AAL3/4 that does not recognise *CryptoTag*, then *CryptoTag* will simply be dropped and we revert to native AAL1 and AAL3/4. This only means the receiver cannot validate the MAC or picks up new IV. Therefore the mechanism will still works for integrity protected stream. This is particular useful in a point-to-multipoint configuration where some of the receiving endpoints are not capable of handling cryptographic operations.

The *CryptoTag* mechanism can be extended in the following way to accommodate larger IV or MAC. By taking advantage of ATM no re-ordering property, we can associate a few *CryptoTags* with a cryptographic unit at a time. We can further use different CPI fields to differentiate the *CryptoTags*.

## 5.5 Encryption of AAL5 frames

We have handled the encryption of AAL5 frames specially. In the previous discussion, the encryption is done on all data in the payload. However the encryption of the AAL5 trailer will cause a conventional ATM host interface to discard the frame due to failed CRC. Therefore for a AAL5 frame, we do not encrypt the AAL5 trailer and the CRC is generated for the encrypted frame.

In the environment, as shown in Figure 9, where a plaintext AAL5 passes through a switched firewall which performs cipher of the AAL5 frame data. In this case, we would like to preserve the state of the CRC, as that indicates the error status of the frame before reaching the firewall. The following technique can be used for efficient determination of the new CRC:

$$CRC' = CRC(\text{Plaintext } XOR \text{ Ciphertext})$$

$$New\ CRC = (\text{OLD CRC})XOR(\text{CRC}')$$

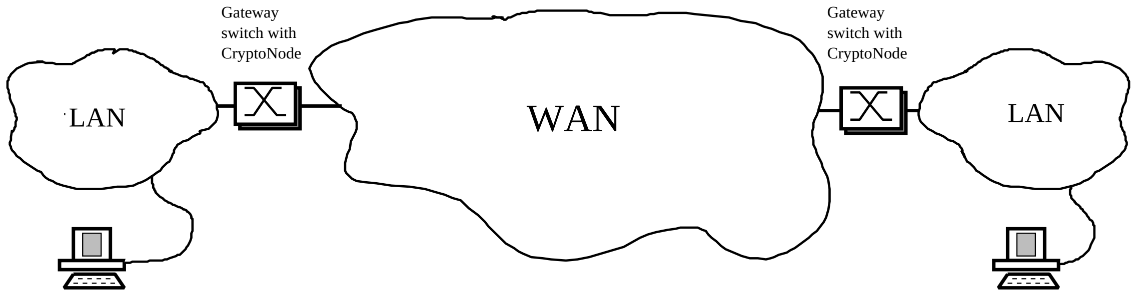


Figure 9: Scenario for AAL5 CRC recalculation

The ability to detection a corrupted CRC of an AAL5 frame enable us to distinguish corruption due to communication error and corruption due to attacker's malicious modification. This distinction is useful for the correct action to be taken to handle the corruption.

### 5.6 Encryption of AAL3/4 frames

The AAL3/4 has the 10 bits *multiplexing identifier (MID)* field which is used for multiplexing different CS-PDUs. If multiple AAL connections use the same ATM layer connection, these AAL connections must have identical QoS characteristics. Since in our model, security profile is part of the QoS parameters, therefore all the different AAL connections use the same ATM layer connection is under the same security protection.

### 5.7 Dealing with Operations and Maintenance (OAM) cell

OAM information flows is structured in a hierarchical manner as in Table 1.

| Level | Flow |
|---|---|
| Regenerator section | F1 |
| Digital section | F2 |
| Transmission path | F3 |
| Virtual path | F4 |
| Virtual channel | F5 |

Table 1: OAM Hierarchical Levels and Flows

Since OAM operations involve inserting OAM cells into user plane data channel, we need to handle them very carefully when the user plane data is processed by the ATM ciphering device such as the *CryptoNode*. For flows F1, F2 and F3 which belong to the physical layer, the *CryptoNode* should recognise and prevent ciphering operations on these cells.

Flows F4 and F5, which belong to the ATM layer, are usually generated/terminated either at the endpoints of a VPC/VCC or at the connecting points terminating a VPC/VCC segment. Since they maybe also optionally be monitored at intermediate points ("in-flight"), the exact ciphering semantic on these OAM cells can be setup either during call setup time or setup in a more permanent basis with network management mechanisms.

The other type of OAM cell that we need to handle specially is the *Resource Management (RM)* cells. These are cell used for congestion control for ABR traffic and they are examined and possibly modified by the intermediate nodes. Therefore these cells should be skipped during the ciphering operation.

Further discussions on the security semantic of OAM cells can be found in Section 7.6 when we look into securing the management plane of the BISDN Reference Model.

## 5.8 Covert Channels

In this section we identify a few covert channels in the user plane. We also proposed the solution to either eliminate the covert channels or at least reduce the bandwidth of these covert channels.

- *CryptoTag* Frequency/Period

  With AAL0 or AAL1 traffic, *CryptoTag* can be inserted at arbitrary point in the data stream. By varying the frequency/period of the *CryptoTag*, one can encode information based on the frequency/period. The obvious way to eliminate this covert channels is to ensure that *CryptoTag* is sent in a periodic fashion.

- PDU Length

  A related covert channels is when dealing with AAL that we protect at PDU boundaries, such as AAL5. The PDU length of the AAL5 unit can be used as the covert channels. We can made the PDU length of to be constant for a particular VC. This length can be negotiated at call setup time. This still leaks some information. However since it is over the time scale of call time, the covert channel bandwidth is greatly reduced. The other option of a absolute constant PDU length for all VC may be considered too restrictive.

- Traffic generation

  Even with the PDU length fixed to a constant value, the frequency/period between the PDU can still be used as the covert channel. To eliminate this, we only allow data to be transmitted at a constant rate, i.e. we made the traffic Constant Bit Rate (CBR). The bandwidth requirement of the VC is set to be the peak data rate. If the data source has no data to send, then artificial PDUs filled with random data are generated. These artificial PDUs must also be indistinguishable from real PDUs after encryption, though the data endpoints can insert marker in the plaintext to indicate the status of each PDU. A timer mechanism can be used to decide when to transmit a PDU. If a PDU with real data is ready for transmission, then it is enqueued and transmitted instead of an artificial PDU the next time the timer expires.

- VCI

  The VCI number of a VC can be used as a covert channel, though in the case of ATM, this is only limited to one hop. This channel appears because some connection management software create a new VCI by incrementing a VCI variable whenever a new VC is created. One can reduce the bandwidth of this channel by changing the VCI allocation algorithm of the TCB connection management software). That is, the TCB could allocate unused VCI in the VCI space (pseudo) randomly in a non-monotonic sequence. Depending upon the randomisation characteristic of the allocation algorithm, the bandwidth of the VCI channel can be reduced to negligible values.

  For some signalling system, such as ATM Forum UNI [20], the VCI is allocated by the initiator. Therefore the initiator will have the control over the covert channel and in some way be able to control the bandwidth.

## 5.9 Implementations

Our implementation work uses the Fairisle Port Controller [21] to simulate the functions of the *CryptoNode*. The Fairisle Port Controller basically consists of the network and the processor section of the *CryptoNode*. Thus in our experiment, we are using the Fairisle Port Controller processor section to perform the dual functions of crypto and processor section of the *CryptoNode*.

Fawn [22], a micro-kernel that is capable of supporting QoS guarantee, runs on the Fairisle Port Controller as the controlling operating system. With such an operating system, we are able to allocate a guaranteed share of the CPU for the handling of the key hint event while processing the cell streams.

We uses DES CBC mode to both encrypt the data for confidentiality and calculating the MAC. In the normal MAC calculation, we would use the last black of the ciphering output as the MAC. However we perform an extra process to increase the strength of the MAC with respect to chosen plaintext attacks. In this process, two cryptographic keys are used, which are denoted by $(K)$ and $(K_1)$, where $(K_1)$ maybe derive from $(K)$[3]. The last block of the ciphering output is encrypted using $(K_1)$ to obtain the final MAC.

[3] An example of how to derive $(K_1)$ from $(K)$ is to complement alternate blocks of four bits of $K_1$ commencing with the first four bits

# 6 Securing the Control Plane

In this section we will examine the security issues involved in the control plane. We first look at our control and management model for the ATM networks. With this model in mind, we then try to look at the fundamental security requirements impose on the the design of signalling system. In particular, we are concerned with the key management protocols for the establishment of the above secure channel. Issues such as access control and the integration with policy routing are discussed. Furthermore, we consider the delegation of authorisation and third party call setup in an Open Multi-Service Network. Firewalling and intrusion detection for ATM are also discussed in this context.

Also in the control plane, we investigate the possible extensions to ATM routing and call setup protocol to achieve Byzantine robustness. Two schemes are considered here, one with an integration with the P-NNI specification [23], while the other call setup scheme uses secure broadcast to achieve Byzantine robustness.

The more detailed discussion and design of these security issues involve in the control plane can be found in [24].

## 6.1 Network Control Model

Our network control model share the same philosophy as the DCAN project [25], of devolving the management and control of switches and end devices into a coherent external system, therefore removing the need for switch and device manufacturers to capture these functions within their equipment. This is illustrated in Figure 10.

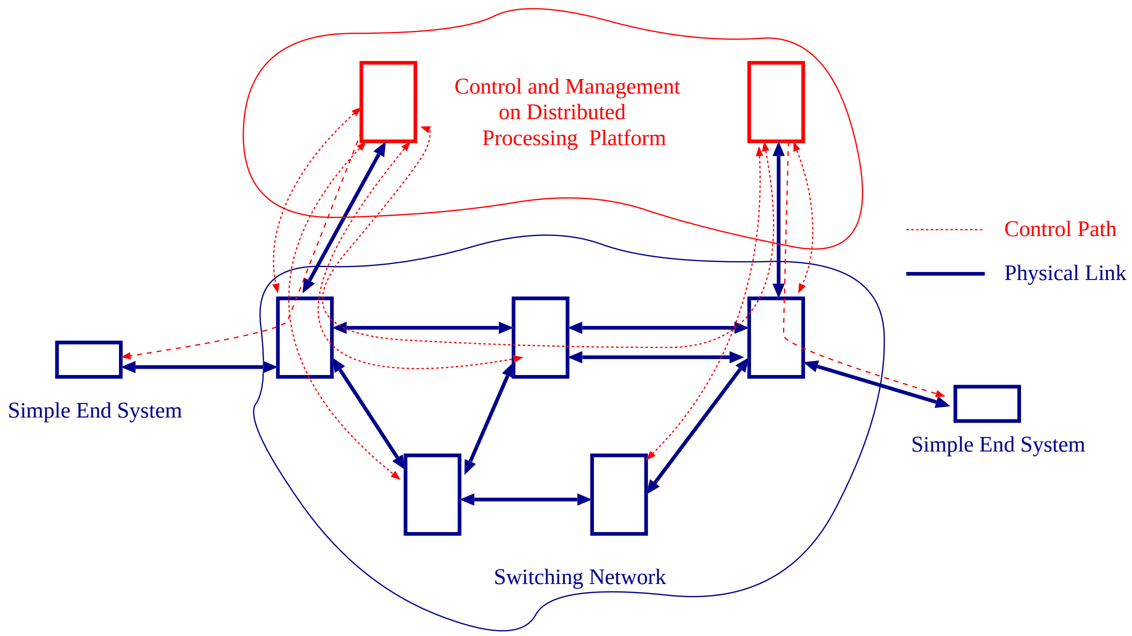


Figure 10: Devolved Management and Control

The control and management of multi-service networks is viewed as a distributed problem and distributed platform such as ANSAware [26].

## 6.2 Signalling

Security issues in signalling can be broadly divided into two categories:

- Security Signalling

  This is the ability in the signalling system to provide the capability to invoke security service provide by the ATM network, such as the data integrity and confidentiality service in the user plane as discussed in Section 5. The ability to provide Byzantine robustness routing and call setup is also one of the services provided by security signalling.

- Signalling Security

  This is the ability to ensure the integrity of the signalling system itself. One of the requirement is that the signalling message can be assured of data origin (source authentication) and integrity. Thus signalling security encompasses authenticated signalling as the pre-requisite. The kind of authentications required include:

  - user to network
  - third party user to remote network
  - inter-switch or inter-network

It is clear from the above discussion that authentication protocol plays a key part in tackling security issues in signalling. We have seen in recent months various proposals of authentication/key management protocols for the establishment of session keys for the secure channel. However, most of these proposals concentrate on finding a particular one (or familiar of) key management protocol to fit in the ATM Forum UNI protocol [20]. In this paper, we are also proposing a number of key management protocols but we emphasis a more fundamental approach in examining the clean integration of such key management protocols with the design of the the ATM signalling protocol.

In [4, 5], compatibility with the current signalling standard and transparency of operation is stressed. The key management protocol is carried out either over the data channel or extra channel setup for such purpose. This is a rather ad hoc approach. The approach is also closely coupled with the particular key management protocol in used and can hence not be flexibly extended to other key management protocols.

We envisage the adaptation of key management protocols to be an evolving process, i.e. improved new key management protocols will replaced old and broken ones as time progress. The underlying signalling protocol must therefore accommodate such changes. We also see that the key management protocol and other security services must be integrated into the ATM service. This is also a view shared in [27]. The strengths in using this approach are:

1. There is no need for any entity involved in providing security service to peek into the user channel.
2. We can prevent the re-invention of all the procedures of using the data channel in the case of key management protocol change.
3. A coherent handling of the key management protocol can not only reduce any unnecessary round trip delay, the protocol processing can also be reduced.
4. Security services can be accessed in a uniform way, just like other ATM services.

We will now examine a few constraints on the key exchange protocol:

- Avoid timestamp

  We should avoid the use of timestamp in the protocol as synchronised clock is a hard thing to get.

- Round trips

  The number of round trip delay for the whole key exchange protocol should be within a limit. If the ATM Forum UNI protocol [20] is used, this is restricted to two-ways. If the MSNL signalling protocol [28] is used, we can extend that to three phases. The round trip delay factor can be particularly bad when an ATM WAN is involved.

- Performance

  This is constraint by the call setup time requirement.

- Cross-domain call

  For calls that are crossing administrative domains, it is desirable to have the property of non-repudiation. This probably imply the use of public key system.

- Uni-directional Key

  Given the key lifetime problem that we have observed, it is desirable to establish a uni-directional key so that keys can be expired independently in each direction.

- Multiple key exchange

  For each run of the key exchange protocol, it is efficient to exchange more than one key at a time. This reduce the need for subsequent re-run of the key exchange protocol. This is also useful when both the confidentiality and integrity key can be established at the same time.

## 6.3 Key Management Protocol

In this section we will examine a few key exchange protocols that satisfy the above constraints.

### 6.3.1 A Nonce-based Challenge/Response Protocol

Notations:
$C_A$ = A's public key certificate
$N_A$ = A's challenge
$E_A(M)$ = Encrypted M with A's public key
$S_A(M)$ = Signed hash of M with A's private key
$K_{AB}$ = Session key for flow from A to B
C = Confounder

The protocol is as shown below:

$$A \rightarrow B : E_B(N_A)$$

$$B \rightarrow A : E_A(N_B), S_B(E_A(B, N_A, K_{BA}, C))$$

$$A \rightarrow B : S_A(E_B(A, N_B, K_{AB}, C1))$$

### 6.3.2 CCITT X.509

The CCITT X.509 authentication protocol [29] can be in either one-way, two-way, or three-way handshake. In both one-way or two-way, a synchronised clock is needed which therefore do not satisfy our constraint. Problems in the X.509 protocol have also been identified in [30]. An improved version, which overcome some of the problem, is as shown below:

$$A \rightarrow B : \{\ A,\ R_A,\ T_A,\ \{K_{AB},\ A\}_{K_B}\ \}_{K_A-1}$$

$B \rightarrow A : \{ \ B, \ R_B, \ T_B, \ \{K_{BA}, \ B\}_{K_A} \ \}_{K_B-1}$

$A \rightarrow B : \{R_B, \ B\}S_A$

### 6.3.3 The Station-to-Station Protocol

This is a protocol suggested by Diffie [31] for use in the Northern Telecom Secure ISDN system. It is based on Diffie-Hellman key establishment [32] follow by an exchange of authentication signature.

Notations:
p = prime
$\alpha$ = a primitive element in prime field GF(p)
$E_A(M)$ = Encrypted M with A's public key
$S_A(M)$ = Signed hash of M with A's private key

The protocol is as shown below:

$A \rightarrow B : \alpha, \ p, \ \alpha^x \ mod \ p$

$B \rightarrow A : \alpha^y, CERT_B, E_K(S_B\{\alpha^y, \ \alpha^x\})$

$A \rightarrow B : E_K(S_A\{ \ \alpha^x, \alpha^y\})$

$where \ CERT_A \ = \ (Alice, \ P_A, \ \alpha, \ p, \ S_T\{Alice, \ P_A, \ \alpha, \ p\})$

The session key is

$K = \alpha^{xy}$

## 6.4 Observation from the above protocols

One observation from the above different key management protocols is that given the constraint of not using synchronised clock, we would need a three-phase handshake in the key management protocol. Thus we are strongly advocating that we also require a three-phase handshake in the corresponding signalling protocol. With a three-phase handshake signalling protocol, we also allows key management protocols involving synchronised clock if such clocks are available.

It is clear that security functional requirements for ATM signalling system should:

- Allow different security policy
- Allow multiple crypto systems
- Allow multiple protocols

Since the ATM Forum UNI protocol is restricted to only two-phase, it seems that it is not a signalling protocol sufficiently generalised to accommodate different key management protocols.

We have based our secure signalling work on the signalling protocol of Multi-Service Network Connection Management Architecture (MSN-CMA) as described in [28]. This is a three-way handshake protocol, designed to avoid the problem of unreliability of the signalling network causing confusion to the state of the connection. In MSN-CMA, an ATM connection is called a liaison. Therefore a secure channel in MSN-CMA is a secure liaison. The signalling protocol is also

RPC-based. This enable us to develop our network control model in the direction of the TINA consortium [33] in which distributed processing environment is used to provide the infrastructure for interaction.

The other observation from the above protocols is that although we have avoided the problem of synchronised clock, we still need to generate good random numbers. For most convention workstations and ATM switches, there is little hardware support for generating good random number.

## 6.5 MSN-CMA Signalling Protocol Security Extension

One extra requirement on the signalling protocol arise from the above work is the need for key renewal signalling part. The functionality that this provides is basically similar to the call setup part with the difference that no further circuit is setup. This design of a key renewal signalling part is also used in the in-call re-negotiation of QoS.

In fact, we are treating the security profile of a secure liaison as part of the QoS parameters and thus negotiated at call setup time. This is similar to the view expressed in [27]

Possible QoS parameters on a liaison are:

- Byzantine robustness call setup (see Section 6.9)
- Covert channel free
  - length: all transmitted frames are of equal length,
  - rate: each transmitter transmits at a constant average rate
  - artificial frames are generated and contain no real data, but are indistinguishable from frames containing real data
- Confidentiality mechanism
  - cryptographic algorithm
  - key size
  - IV
- Integrity mechanism
  - cryptographic algorithm
  - key size
- Key exchange mechanism
- CryptoTag policy
  - key period (if periodic mode or keep-alive mode)
  - key lifetime
- Stream, frame or covert channel free mode

The MSN-CMA operates in terms of MSNL Management Domain (MD). A MD is a collection of one or more network entities grouped together for the purposes of management and control. This can also be interpreted in the security sense as under the same security policy. The MSNL Master is responsible for co-ordinating the management of communication resources within the MD and also enforcing the security policy. The User Master Interface is used to signal user requirements to the MSNL Master and the Master Master Interface is used for inter-Master signalling. There is also the Slave Control Interface which allows the MSNL Master to construct and destroy data paths within an entity. All these interfaces are defined in an Interface Definition Language (IDL) and the procedure abstraction is supported by a RPC mechanism. Therefore one of the first step to achieve signalling security is to provide security in the underlying Remote Procedure Call (RPC) mechanism.

Most of the IDLs are unable to express the semantic and level of protection (integrity or confidentiality) of the interface. Proposal to have a more expressive extensions to IDL and the RPC system and also a detailed discussion of security enhanced MSN-CMA protocol (with point-to-multipoint considerations) can be found in [24].

The other infrastructure consideration is the Certification Authority structure required for public key system. Much of these is generic to other systems. One factor that might influence the choice of Certification Authority mechanism is the short time scale of operation in the control plane. This might affects decision such as whether using pull or push model of handling certificate. Certificate revocation can also make a performance impact. Detailed discussion can be found in [24].

## 6.6 NNI Routing and Inter-Domain Policy Routing

The Network Node Interface (NNI) is the interface between network nodes. The ATM Forum has an ongoing effort to define a Private Network Node Interface (P-NNI) protocol for use within networks that use NSAP format ATM address. One of the component of the P-NNI is a virtual circuit routing protocol. The protocol uses a topology state routing protocol in which node flood QoS and reachability information so that all nodes obtain knowledge about reachability within the network and the available traffic resources within the network. Such information is passed within P-NNI topology state packets (PTSP), similar to current link state routing protocol such as OSPF.

The P-NNI is scalable with a hierarchical network organisation, with summarisation of reachability information between levels in the hierarchy.

The P-NNI signalling protocol performs source routing by the source network node. A designated transit list (DTL) that described the complete route to the destination is inserted into the signalling request. This source routing approach can be integrated very well with the Inter-Domain Policy Routing (IDPR) [34] to provide network policy support for access control.

Much of the mechanisms in IDPR has to do with the setting up of the policy route using the *SETUP* packet and retaining of state in the router. However, for ATM, most of these complications has been taken care of by the NNI signalling protocol. Thus we can have a very clean integrate the concept of IDPR for access control into ATM network P-NNI and results in a simpler system.

In IDPR [34], the *Visa* protocol is used for stub network policy enforcement and special route setup scheme and cryptographic signature to handle transit internetwork traffic. The cryptographic measure is to ensure that network resources are not been "hijacked" by illegal traffic piggy-backing on a legal policy route by discarding illegal packets at the earliest stage.

At ATM call setup stage, we can adopt a similar *Visa* protocol scheme for stub network and route setup scheme for transit network. However, the access control on data might prove infeasible. ATM network is cell relay and data integrity checks at cell level simply involve too much overhead. A straight adaptation of the IDPR approach would turn the border policy switch into a frame relay which is highly unsatisfactory.

## 6.7 Authorisation and delegation

The above discussion on policy routing touches on the issues of network resource usage control. In an Open Multi Service Network built upon ATM networks, an Open Service Support Architecture enables easy provision of services. In such an environment, a user must be able to obtain services as needed in as transparent a mode as possible, while a wide range of authorisation and billing services are used underneath. Thus this enables third party service providers to use resources on the network that their client have authorisation (and delegation powers) to use. Such facility will also allow functions such as secure third party call setup. Auditable delegation of authority is also desired. The *HATS* project in the computer laboratory is looking into this problem.

## 6.8 Firewalling and Intrusion Detection

Unlike authentication and privacy mechanisms, which can be added to application programs, network access control usually requires changes to basic components of the network infrastructure. For the internet, we need significant processing power on the firewall since the firewall needs to examine each datagram that goes between the internet and the external part of the internet. For ATM networks, firewalling can be done at the call setup time. For high speed network, it is almost impossible to have enough processing power to examine all the user data. But for a connection oriented protocol with a clear separation of control plane from the user plane, such as ATM, the access control can be done during connection setup time. The fine granularity of ATM cells also ensure low delay through a firewall. It has been proposed [35] to include higher layer application authenticator as an Information Element in the signalling message. We do however rely on the trust on the higher layer application to not generate any malicious data.

Also due to the non-broadcast nature (mostly) of ATM and the high data volume, intrusion detection mechanisms that rely on passive monitoring of network will not work on ATM network. This push intrusion detection system for ATM to be limited to host-based.

## 6.9 Byzantine Robustness Routing and Call Setup

In order to prevent denial of service attack, we try to design an ATM network that will guarantee that a call setup between two non-faulty end systems will have a high probability of being delivered, provided that at least one path consisting of non-faulty components connects the two end system. In this section, we outline two schemes that achieve this.

### 6.9.1 Robust P-NNI

The P-NNI routing protocol floods P-NNI topology state packets (PTSP) in a manner similar to current flooding of link state packet for protocol such as OSPF. Therefore techniques that achieved robust flooding for link state packet [36] can be modified to handle robust PTSP flooding.

We need to firstly use a public key distributor to distribute the public keys of all the nodes. The public key distributor flood the public key list packet.

A buffer for a packet is reserved for each source (network node) at each switch node. A digital signature is put into the packet to ensure that only the packet generated by the source occupies the buffer. The link state packet is then flooded.

Now the switch node should be able to compute a route, in the normal manner that P-NNI protocol does and construct a DTL.

Hierarchical networks as is possible with P-NNI can also be handled as described in [36].

### 6.9.2 Robust Call Setup

The other more radical approach is to deliver the call setup signalling message using robust flooding. Key to robust flooding is guaranteed resource for each source. Therefore at each of the switch node, we need to reserve buffer for the call setup signalling message. The call setup throughput, i.e. the number of concurrent call setups possible is determined by the number such buffer set aside.

It might seem that such flooding is rather resource wasteful. However, we only carry out such flooding once in the lifetime of a call, i.e. at the call setup time. Whether such mechanism is worthwhile also determine by the call holding time and the bandwidth requirement of each call. The selection of this kind of call setup versus a more normal approach should be specified in the QoS parameter. However, a re-dimensioning of the signalling channel might be needed given the extra signalling traffic.

The detailed mapping of the mechanism to the P-NNI protocol and the Robust Call Setup scheme can be found in [37].

# 7 Management Plane Security

The ATM network management framework is not fully developed. Therefore we based our discussions in this section on some of the current development.

## 7.1 Start of the Day - Bootstrapping

The secure bootstrapping process of ATM switches is crucial to the later security operations. Yet it is one of the area that has not been investigated in much detail. This is particularly important to the provision of open multi service networks. There is an increasing need for booting ATM switches with third party software, using a third party booting service. [38] present a case study of secure ATM switch booting based on the Fairisle ATM network.

In this work, when the switch is finally booted, we can be assured that it has been loaded with the correct software and it has acquired a new public key pair.

## 7.2 Authenticated Neighbour Discovery

When a network node is being bootstrapped, it does not have reachability information about who it is connected to. The information is gathered using some kind of *Hello* protocol across some PVCs. For protocol such as P-NNI, *Hello* packets are exchanged on a well known virtual channel. Such *Hello* packets should carry an authenticator which has the following information:

- Sender Address
- Timestamp
- Sequence number/nonce
- Signature/Hash Algorithm
- Digital Signature

The timestamp and sequence number/nonce are used to prevent replay attack. However, accurate timestamp is hard to come by, especially at bootstrap stage. Therefore one possible improvement over the above is to remove the use of timestamp and turn the *Hello* protocol into a challenge/response protocol.

## 7.3 ILMI Security

The Interim Local Management Interface (ILMI) is a temporary solution to providing some kind of management capability. The UNI Management Entity (UME) response to SNMP command received over the ATM UNI through the ILMI. In order to give the ILMI some security functionality, it can be enhanced with the SNMP security protocol [39]. An initial key exchange stage should also be added using the ILMI PVC to establish mutual authentication and generate session key for the secure SNMP session. With this enhancement, secure address registration can also be done through the ILMI.

## 7.4 Permanent Virtual Circuit security

Permanent virtual circuit (PVC) is supposed to be setup using some network management mechanism. It is highly desirable that PVC will have the same security protection as the secure liaison we have discussed earlier. Therefore, the network management mechanism must allow the security profile of a PVC to be setup. If the network management mechanism is through something similar to SNMP or CMIP, then the security profile should be defined as objects in the MIB.

### 7.5 VPI Security

Virtual path (VP) is seen as a bundle of VCs setup on a longer timescale basis. The setup of VP can either be through signalling or network management. Since VP can be viewed as a "fat" VC, most of the mechanism we have discussed relating to VC also apply to VP. A VP between users provides them with a transmission 'pipe', the VC organisation of which is up to them. This concept maybe applied to LAN-LAN coupling. If this LAN-LAN coupling is across some untrusted WAN, for example, then it would need protection across the WAN and it would make sense have all the protection on the VP. Since the matter of protecting VC or VP is highly dependent on usage and even network configuration, it is vital that the mechanism to setup a secure channel enables the choice to be made on the kind of protection required.

### 7.6 ATM Forum Management Model

The ATM Forum is developing a 5 layer Management Model and the OAM facility. ILMI, the temporary model has been discussed before. The 5 layer management interfaces are labelled M1 through M5. M1 and M2 embrace SNMP-based specification. It is expected that much experience in securing SNMP and CMIP can be brought forward to securing the M1-M5 management protocol.

The ATM Forum has also specified a few OAM cells targeting fault management. The question of whether these OAM cells require any security protection is debatable. For this moment we assume such need does exist. It is quite unlikely that there is a need to encrypt the cell payload. However, since these are single cells message, cryptographic operation to provide any integrity or authenticity is quite impossible. However, we can probably adopt the *CryptoTag* mechanism to circumvent the problem. Let's consider a VP or VC interrupted due to switch failure. This will generate AIS (alarm indication signal) in the downstream. This AIS cell is neither authenticated or integrity protected. We have talked about the possibility of an extended *CryptoTag* mechanism which consist of a serial of *CryptoTags*. A public key digital signature could be 'stripped' across the payload of these *CryptoTags*. Similar operation can also be applied to other OAM cells such as FERF (far end reporting failure).

## 8 Higher Level Services

The impact of the above discussed ATM Network security measures on higher level services such as LAN Emulation and IP over ATM will be reported in the next release of this paper.

## 9 Conclusion

We have considered a wide range of issues in securing the ATM networks. We have covered all the three plane of the BISDN reference model. The techniques to achieve data confidentiality and data integrity for a secure ATM channel and the corresponding techniques to establish session keys for such channel were proposed. Key agile cryptographic device for ATM is being described.

Other security issues such as secure switch booting, fault-tolerance call setup, call authorisation, secure ATM network management and secure signalling etc have all been considered. This list is by no mean exhaustive and there are much work to further study these issues. Future revisions of this document will revisit some of the issues covered in this paper and also attempt to address new areas of security risk in ATM networks.

# References


[1] Ian M. Leslie, Derek M. McAuley and David L. Tennenhouse, "ATM Everywhere ?," *IEEE Networks*, March 1993.

[2] I. Pratt and P. Barham, "The ATM Camera V2: AVA200," in *ATM Document Collection 3 (The* Blue *Book)*, ch. 36, University of Cambridge Computer Laboratory, March 1994.

[3] M. D. Hayter and D. R. McAuley, "The Desk Area Network," *ACM Operating Systems Review*, vol. 25, pp. 14–21, October 1991.

[4] D. Stevenson, N. Hillery, and G. Byrd, "Secure Communications in ATM Networks," *Communications of ACM*, vol. 38, pp. 45–52, February 1995.

[5] D. Stevenson, N. Hillery, F. G. Grep Byrd, and D. Winkelstein, "Design of a Key Agile Cryptographic System for OC-12c Rate ATM," in *Symposium on Network and Distributed System Security*, pp. 17–30, February 1995.

[6] Cheswick W., and Bellovin S., *Firewall and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, Reading, Mass, 1994.

[7] D. Tennenhouse, "Layered multiplexing considered harmful," in *IFIP WG6.1/6.4 Workshop: Protocols for High Speed Networks, IBM Zurich Research Lab.* (H. Rubin and R. Williamson, eds.), May 1989.

[8] H. Eberle, "A High-Speed DES Implementation for Network Applications," *Advances in Cryptology - CRYPTO 92, Berlin:Springer-Verlay*, pp. 527–545, 1993.

[9] A. S. R. Rivest and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, pp. 120–126, February 1978.

[10] Korbus van der Merwe and Shaw-Cheng Chuang, "Support for Open Multi Service Networks," in *International Teletraffic Seminar on Teletraffic Engineering for Evolving Networks, Pretoria, South Africa*, To appear September 1995.

[11] Korbus van der Merwe and Shaw-Cheng Chuang, "Support for Open Multi Service Networks," in *International Teletraffic Seminar on Teletraffic Engineering for Evolving Networks, Pretoria, South Africa*, September 1995.

[12] D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," *ACM SigComm*, September 1990.

[13] S. Mullender, I. Leslie, and D. McAuley, "Operating-System Support for Distributed Multimedia," in *USENIX Summer 1994 Conference*, July 1994.

[14] X/Open Company Ltd, *X/Open Generic Cryptographic Service API (GCS-API) Draft 2*, February 1995.

[15] J. Wray, "Generic Serurity Service API : C-bindings," September 1993.

[16] National Institute of Standards and Technology, *NIST Cryptographic Service Calls Draft*, May 1994.

[17] I. Pratt, "Hardware Support for Operating System Support for Continuous Media." University of Cambridge Computer Laboratory Ph.D. Research Proposal, July 1994.

[18] I. Pratt, "Memory System Requirements for Multimedia Workstations." University of Cambridge Computer Laboratory Ph.D. Discussion Paper, February 1995.

[19] A. Romanow and S. Floyd, "The Dynamics of TCP Traffic over ATM Networks," in *Computer Communication Review*, vol. 24, pp. 24–35, ACM SIGCOMM, September 1994.

[20] ATM Forum, *ATM Forum User Network Interface Specification V3.0.* Prentice Hall, 1994.

[21] I. Leslie and D. McAuley, "Fairisle: An ATM Network for the Local Area," in *Computer Communication Review*, vol. 21(4), pp. 327–336, ACM SIGCOMM, September 1991.

[22] R. J. Black, *Explicit Network Scheduling.* PhD thesis, University of Cambridge, December 1994.

[23] ATM Forum, "ATM Forum 94-0471R6, P-NNI Draft Specification," 1994.

[24] Shaw-Cheng Chuang, "Security-related Signalling," tech. rep., University of Cambridge Computer Laboratory - In preparation, June 1995.

[25] Ian Leslie, "Case for Support: Scalable Distributed Control of ATM Networks," July 1994.

[26] Architecture Projects Management Limited, *ANSAware/RT 1.0 Manual*, 1995.

[27] L. G. Pierson and T. D. Tarman, "Requirements for Security Signalling," *ATM Forum Contribution*, February 1995.

[28] S. A. Crosby, *Performance Management in ATM Networks.* PhD thesis, University of Cambridge, May 1995.

[29] CCITT, *The Directory-Authentication Framework*, 1989. Blue Book - Melbourne 1988, Fascicle VIII.8: Data Communication Networks: Directory, International Telecommunications Union, Geneva, Switzerland pp. 127-141.

[30] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," tech. rep., Digital Equipment Corp. System Research Center, February 1990.

[31] W. Diffie, P. C. V. Oorschot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges," *Designs, Code and Cryptography*, no. 2, pp. 107–125, 1992.

[32] Diffie W. and Hellman H. E., "New Directions in Cryptography," *IEEE Trans. Info. Theory*, no. IT22-(6), pp. 644–654, 1976.

[33] T. B. William J Barr and Y. Inoue, "The TINA Initiative," *IEEE Communications Magazine*, March 1993.

[34] G. Tsudik, *Access Control and Policy Enforcement in Internetworks.* PhD thesis, University of California, April 1991.

[35] Bryan Lyles, "Authenticated Signalling: T1S1.5/94-118," 1994.

[36] R. Perlman, *Network Layer Protocols with Byzantine Robustness.* PhD thesis, MIT LCS TR-429, December 1983.

[37] Shaw-Cheng Chuang, "Call Setup and Routing Scheme with Byzantine Robustness," tech. rep., University of Cambridge Computer Laboratory - In preparation, July 1995.

[38] Shaw-Cheng Chuang and Michael Roe, "A Case Study of Secure ATM Switch Booting," in *preparation for submission to The Internet Society Symposium on Network and Distributed System Security*, February 1996.

[39] J. Galvin, M. McCloghire and J. Davin, "SNMP Security Protocol," July 1992.